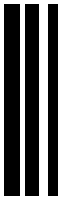

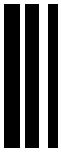

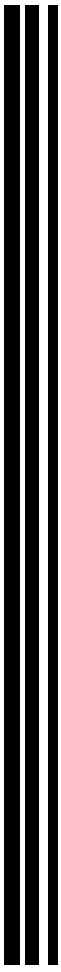

SUMMARY TABLE FOR: X.509 Internet Public Key Infrastructure Online Certificate Status Protocol (OCSP) March 1998

22 June 1998

Prepared by:

Center for Standards
Defense Information Systems Agency

This supercedes version dated DD MMM YYYY and all earlier versions.

This Page left intentionally blank.

Disclaimer

Persons and organizations use this document at their own risk.

This document is for information only. If there is any conflict between this document and the source document, the source document takes precedence.

The U. S. Federal Government does NOT provide any guarantee as to the accuracy of this document. This document is NOT a request for proposal, a request for bid, or a modification to any contract currently held with the U. S. Federal Government.

Distribution of this document is unlimited.

Acronyms

This Page left intentionally blank.

STATUS CODES: M – MANDATORY, O – OPTIONAL, C - CONDITIONAL

SECTION	FEATURE	STATUS	REMARKS
2	**Protocol Overview**		
2.1	**Request**		
	An OCSP request contains the following data:	M	
	protocol version	M	
	service request	M	
	target certificate identifier or a single-entity certificate	M	
	optional extensions which MAY be processed by the OCSP Responder	O	
	Upon receipt of a request, an OCSP responder determines if: 1) the message is well formed, 2) the responder is configured to provide the requested service, and 3) the responder can perform the requested service for the subject certificate. If any one of the prior conditions are not met, the OCSP responder produces an error message; otherwise, it returns a definitive response.	M	
2.2	**Response**		
	All definitive response messages SHALL be digitally signed	M	
	Key used to sign the response MUST belong to one of the following:	M	
	the CA who issued the certificate in question	M	
	a Trusted Responder whose public key is trusted by the requester	M	
	A definitive response message is composed of:	M	
	response type identifier (to allow for different response types)	M	
	version of the response	M	
	name of the responder	M	
	responses for each of the certificates in a request	M	
	optional extensions	M	
	signature algorithm OID	M	
	signature computed across hash of the response	M	
	The response for each of the certificates in a request consists of:	M	
	target certificate identifier	M	
	certificate status value	M	
	response validity interval	M	
	optional extensions	M	
	This specification defines the following definitive response indicators for use in the certificate status value:	M	

SECTION	FEATURE	STATUS	REMARKS
	notRevoked	M	This state indicates that the certificate is not revoked. It does not necessarily mean that the certificate was ever issued. Nor does it mean that the certificate is in its validity interval. A notRevoked state by an OCSP responder DOES NOT absolve the application of the responsibility of checking that the certificate is in its validity period and has been correctly signed.
	revoked	M	This state indicates that the certificate has been revoked.
	onHold	M	This state corresponds to valid certificates that are operationally suspended in accordance with PKIX Part 1.
	expired	M	A request that returns an expired state indicates that the validity of the subject certificate has expired. Applications SHOULD check the validity interval of a certificate and not perform an OCSP request if the certificate's validity has expired.
2.3	**Exception Cases**		
	In case of errors, the OCSP responder may return an error message. Errors can be of the following types:	M	
	malformedRequest	M	A server produces this response if the request received does not conform to the OCSP syntax.
	internalError	M	This response indicates that the OCSP responder reached an inconsistent internal state. The query should be retried, potentially with another responder.
	tryLater	M	In the event that the OCSP responder is operational, but unable to return a status for the requested certificate, the tryLater response can be used to indicate that the service exists, but it temporarily unable to respond.

SECTION	FEATURE	STATUS	REMARKS
	notFound	M	A recipient of a request may not be able to resolve a reference to the subject certificate; a value of notFound is returned in such a case. This value should not be taken as confirmation of the certificate's existence.
	certRequired	M	This response is returned in cases where the server requires the client to supply the certificate data itself in order to construct a response.
	noCRL	M	An extension is defined to enable delivery of CRLs with OCSP responses. However, there is no requirement to list certificates on a CRL in order to use OCSP to acquire revocation status on those certifcates. The error value noCRL is defined for this instance.
2.4	**Response Pre-production**		
	The response validity interval noted in the prior section is composed of a {thisUpdate, nextUpdate} pair of elements in the response syntax.	M	Section 4.2 provides details of the response syntax.
	OCSP responders MAY pre-produce signed responses specifying the current status of certificates at the time the response was produced.	O	
	The time at which the response was produced SHALL be reflected in thisUpdate field of the response.	M	
	If responses are pre-produced, then for a given certificate, the periodicity of this pre-production SHOULD match the response validity interval of the most recently produced response.	O	Recommended
	The time at which the response was known to be correct SHALL be specified in the producedAt field of the response.	M	
	The producer of the response MAY include a value for the nextUpdate.	O	
3	**Functional Requirements**		
3.1	**Certificate Content**		
	In order to convey to OCSP clients a well-known point of information access, CAs SHALL provide the capability to include the AuthorityInfoAccess extension (defined in PKIX Part 1, section 4.2.2.1) in certificates that can be checked using OCSP.	M	
	Alternatively, the accessLocation for the OCSP provider may be configured locally at the OCSP client.	O	

SECTION	FEATURE	STATUS	REMARKS
	CAs that support an OCSP service, either hosted locally or provided by an Authorized Responder, MAY provide a value for a uniformedResourceIndicator (URI) accessLocation and the OID value id-ad-ocsp for the accessMethod in the AccessDescription SEQUENCE.	M	
3.3 ???	**Error Responses**		
	Upon receipt of a request, which fails to parse, the receiving OCSP responder SHALL respond with an error message.	M	
	Error responses MAY be signed.	O	
3.5???	**Signed Response Acceptance Requirements**	M	
	Prior to accepting a signed response as valid, OCSP clients SHALL confirm that:	M	
	the certificate identified in a received response corresponds to that which was identified in the corresponding request.	M	
	the signature on the response is valid.	M	
	the identity of the signer matches the intended recipient of the request.	M	
4	**Detailed Protocol**		
	For signature calculation, the data to be signed is encoded using the ASN.1 distinguished encoding rules (DER) [X.690]	M	
	ASN.1 EXPLICIT tagging is used as a default unless specified otherwise.	M	
4.1	**Request Syntax**		
	OCSPRequest ::= SEQUENCE {	M	
	version [0] EXPLICIT Version DEFAULT v1,	M	
	hashAlgorithm AlgorithmIdentifier,	M	
	requestList SEQUENCE OF Request,	M	
	requestExtensions [1] EXPLICIT Extensions OPTIONAL}	O	
	Version ::= INTEGER { v1(0) }	M	
	Request ::= CHOICE {	M	
	certID [0] EXPLICIT CertID,	M	
	cert [1] EXPLICIT Certificate }	M	
	CertID ::= SEQUENCE {	M	
	issuerNameAndKeyHash Hash,	M	
	serialNumber CertificateSerialNumber}	M	
	IssuerNameAndKey ::= SEQUENCE {	M	
	issuer Name,	M	
	issuerPublicKey SubjectPublicKeyInfo }	M	
	Hash ::= OCTET STRING –hash of IssuerNameAndKey--	M	
3.2	**Response Syntax**		
3.2.1	**ASN.1 Specification of the OCSP Response**		
	An OCSP response at a minimum consists of a responseStatus field indicating the processing status of the prior request. If the value of responseStatus is one of the error conditions, responseBytes are not set.	M	
	OCSPResponse ::= SEQUENCE {	M	

SECTION	FEATURE	STATUS	REMARKS
	responseStatus OCSPResponseStatus,	M	
	responseBytes [0] EXPLICIT ResponseBytes OPTIONAL }	O	
	OCSPResponseStatus ::= ENUMERATED {	M	
	successful (0),	M	
	malformedRequest (1),	M	
	internalError (2),	M	
	tryLater (3),	M	
	notFound (4),	M	
	certRequired (5) }	M	
3.2.1.1	**BasicResponse**		
	ResponseBytes ::= SEQUENCE {	M	
	responseType OBJECT IDENTIFIER,	M	
	response OCTET STRING }	M	
	For a basic OCSP responder, responseType will be id-pkix-ocsp-basic, where:	M	
	id-pkix-ocsp OBJECT IDENTIFIER ::= {id-ad-ocsp}	M	
	id-pkix-ocsp-basic OBJECT IDENTIFIER ::= {id-pkix-ocsp 1}	M	
	OCSP responders SHALL be capable of recognizing and responding to the id-pkix-ocsp-basic response type.	M	
	OCSP clients SHALL be capable of receiving the id-pkix-ocsp-basic response type.	M	
	The value for response SHALL be the DER encoding of BasicOCSPResponse:	M	
	BasicOCSPResponse ::= SEQUENCE {	M	
	tbsResponseData ResponseData,	M	
	signatureAlgorithm AlgorithmIdentifier,	M	
	signature BIT STRING,	M	
	certs [1] EXPLICIT SEQUENCE OF Certificate OPTIONAL	O	
	The value for signature SHALL be computed on the hash of the DER encoding ResponseData.	M	
3.2.1.2	**ResponseData**		
	ResponseData ::= SEQUENCE {	M	
	version [0] EXPLICIT Version DEFAULT v1,	M	
	responderID ResponderID,	M	
	responses SEQUENCE OF SingleResponse,	M	
	responseExtensions [1] EXPLICIT Extensions OPTIONAL }	O	
	ResponderID ::= CHOICE {	M	
	byName [0] Name,	M	
	byKey [1] KeyHash }	M	
	KeyHash ::= KeyIdentifier	M	
3.2.1.3	**SingleResponse**	M	
	SingleResponse ::= SEQUENCE {	M	
	request Request,	M	

SECTION	FEATURE	STATUS	REMARKS
	certStatus CertStatus,	M	
	producedAt GeneralizedTime,	M	
	nextUpdate [0] EXPLICIT GeneralizedTime OPTIONAL,	O	
	singleExtensions [2] EXPLICIT Extensions OPTIONAL }	O	
	CertStatus ::= CHOICE {	M	
	certStatusType [0] EXPLICIT CertStatusType (notRevoked | onHold),	M	
	statusWithTime [1] EXPLICT StatusWithTime }	M	
	StatusWithTime ::= SEQUENCE {	M	
	certStatusType CertStatusType (revoked),	M	
	time GeneralizedTime }	M	
	CertStatusType ::= ENUMERATED {	M	
	notRevoked (0),	M	
	revoked (1),	M	
	onHold (2),	M	
	expired (3) }	M	
	Applications SHOULD determine by observation of the certificate's validity interval that a certificate is expired.	O	Recommended
3.2.2	**Notes on OCSP Responses**		
	If the certStatusType is revoked, onHold or expired, the time field of statusWithTime is the time of revocation, suspension or expiration respectively.	M	
	The date returned for expiration should match the notAfter date of the certificate's validity interval.	M	
	Responses whose nextUpdate value is earlier than the local system time value SHOULD be considered unreliable.	O	Recommended
	Responses whose thusUpdate time is earlier than the local system time SHOULD be considered unreliable.	O	Recommended
3.3	**Mandatory and Optional Cryptographic Algorithms**	M	
	Clients that request OCSP services SHALL be capable of processing responses signed used DSA keys identified by the DSA sig-ald-oid specified in section 7.2.2 of PKIX Part 1.	M	
	Clients SHOULD be capable of processing RSA signatures as specified in section 7.2.1 of PKIX Part 1.	O	Recommended
	OCSP responders SHALL support the SHA1 hash algorithm.	M	
3.4	**Extensions**	O	
	Support for all extensions is OPTIONAL.	O	
3.4.1	**Nonce**	O	The nonce cryptographically binds a request and a response to prevent replay attacks.
	In both the request and the response, the nonce will be identified by the object identifier id-pkix-ocsp-nonce, while the extnValue is the value of the nonce.	M	
	id-pkix-ocsp-nonce OBJECT IDENTIFIER ::= { id-pkix-ocsp 2 }	M	

SECTION	FEATURE	STATUS	REMARKS
3.4.2	**Signed Requests**	O	This extension allows the requester to sign a request.
	The requestor includes an extension that has the signatureIdentifier, the actual bits of the signature and a sequence of certificates to allow the OCSP respoder to verify the signature.	M	
	The data to be signed is just the basic request (none of the extensions).	M	
	The OCSP Responder can verify the singature, potentially using certifcates that have been included with the extension.	M	
	The signature on a request will be identified by id-pkix-ocsp-signature, while the value will be SignatureData.	M	
	id-pkix-ocsp-signature OBJECT IDENTIFIER ::= { is-pkix-ocsp 5 }	M	
	SignatureData ::= SEQUENCE {	M	
	signatureAlgorithm AlgorithmIdentifier,	M	
	signature BIT STRING,	M	
	certs [0] EXPLICIT SEQUENCE OF Certificate OPTIONAL	O	
3.4.3	**CRL References**	O	
	These extensions will be specified as singleExtensions.	M	
	The identifier for this extension will be id-pkix-ocsp-crl, while the value will be CrlID.	M	
	id-pkix-ocsp-crl OBJECT IDENTIFIER ::= { id-pkix-ocsp 4 }	M	
	CrlID ::= SEQUENCE {	M	
	crlUrl [0] EXPLICIT IA5String OPTIONAL,	O	the URL at which the CRL is available
	crlNum [1] EXPLICIT INTEGER OPTIONAL,	O	the CRL number extension of the relevant CRL
	crlTime [2] EXPLICIT GeneralizedTime OPTIONAL }	O	the time at which the relevant CRL was created
3.4.4	**Acceptable Response Types**		
	An OCSP client MAY wish to specify the kinds of response types it understands.	O	
	To do so, it SHOULD use an extension with the OID id-pkix-ocsp-response, and the value AcceptableResponses.	O	Recommended
	The OIDs included in AcceptableResponses are the OIDs of the various response types this client can accept (e.g., id-pkix-ocsp-basic).	M	
	id-pkix-ocsp-response OBJECT IDENTIFIER ::= { id-pkix-ocsp 3 }	M	
	AcceptableResponses ::= SEQUENCE OF { id OBJECT IDENTIFIER }	M	
	OCSP responders SHALL be capable of recognizing and responding to the id-pkix-ocsp-basic response type.	M	Noted in Section 3.3
	OCSP clients SHALL be capable of receiving and processing the id-pkix-ocsp-basic response type.	M	
3.4.5	**Other Extensions**	O	

SECTION	FEATURE	STATUS	REMARKS
	CRL Entry Extensions are also supported as singleExtensions.	M	Specified in Section 5.3 of PKIX part I
4	**Security Considerations**		
App. A			
A.1	**OCSP over HTTP**		
A.1.1	**Request**		
	An OCSP request is an HTTP 1.0 POST method.	M	
	The Content-Type header has the value “application/ocsp-request” while the body of the message is the DER encoding of the OCSPRequest.	M	
A.1.2	**Response**		
	An HTTP-based OCSP response is composed of the appropriate HTTP headers, followed by the DER encoding of the OCSPResponse.	M	
	The Content-Type header has the value “application/ocsp-response”.	M	
	The Content-Length header SHOULD specify the length of the response.	O	Recommended
	Other HTTP headers MAY be present and MAY be ignored if not understood by the requestor.	O	
App B	**OCSP in ASN.1**		
	OCSPRequest ::= SEQUENCE {	M	
	version [0] EXPLICIT Version DEFAULT v1,	M	
	hashAlgorithm AlgorithmIdentifier,	M	
	requestList SEQUENCE OF Request,	M	
	requestExtensions [1] EXPLICIT Extensions OPTIONAL }	O	
	Version ::= INTEGER { v1(0) }	M	
	Request ::= CHOICE {	M	
	certID [0] EXPLICIT CertID,	M	
	cert [1] EXPLICIT Certificate }	M	
	CertID ::= SEQUENCE {	M	
	issuerNameAndKeyHash Hash,	M	
	serialNumber CertificateSerialNumber}	M	
	IssuerNameAndKey ::= SEQUENCE {	M	
	issuer Name,	M	
	issuerPublicKey SubjectPublicKeyInfo }	M	
	Hash ::= OCTET STRING	M	hash of IssuerNameAndKey
	OCSPResponse ::= SEQUENCE {	M	
	responseStatus OCSPResponseStatus,	M	
	responseBytes [0] EXPLICIT ResponseBytes OPTIONAL }	O	
	OCSPResponseStatus ::= ENUMERATED {	M	
	successful (0),	M	Response has valid confirmations
	malformedRequest (1),	M	Illegal confirmation request
	internalError (2),	M	Internal error in issuer
	tryLater (3),	M	Try again later
	notFound (4),	M	Certificate not on record
	certRequired (5) }	M	Must supply certificate
	BasicOCSPResponse ::= SEQUENCE {	M	

SECTION	FEATURE	STATUS	REMARKS
	tbsResponseData ResponseData,	M	
	signatureAlgorithm AlgorithmIdentifier,	M	
	signature BIT STRING,	M	
	certs [1] EXPLICIT SEQUENCE OF Certificate OPTIONAL }	O	
	ResponseData ::= SEQUENCE {	M	
	version [0] EXPLICIT Version DEFAULT v1,	M	
	responderID ResponderID	M	
	responses SEQUENCE OF SingleResponse,	M	
	responseExtensions [1] EXPLICIT Extensions OPTIONAL }	O	
	ResponderID ::= CHOICE	M	
	byName [0] Name,	M	
	byKey [1] KeyHash }	M	
	KeyHash ::= KeyIdentifier	M	SHA-1 hash as defined in PKIX Part 1
	SingleResponse ::= SEQUENCE {	M	
	request Request,	M	
	certStatus CertStatus,	M	
	producedAt GeneralizedTime,	M	
	nextUpdate [0] EXPLICIT GeneralizedTime OPTIONAL,	O	
	singleExtesnions [2] EXPLICIT Extensions OPTIONAL }	O	
	CertStatus ::= CHOICE {	M	
	certStatusType [0] EXPLICIT CertStatusType (notRevoked | onHold),	M	
	statusWithTime [1] EXPLICIT StatusWithTime }	M	
	StatusWithTime ::= SEQUENCE {	M	
	certStatusType CertStatusType (revoked),	M	
	time GeneralizedTime }	M	
	CertStatusType ::= ENUMERATED {	M	
	notRevoked (0),	M	This serial number is not revoked
	revoked (1),	M	Serial number was revoked
	onHold (2),	M	Cert is on hold
	expired (3) }	M	Certificate is expired
	-- Extensions		
	SignatureData ::= SEQUENCE {	O	
	signatureAlgorithm AlgorithmIdentifier,	M	
	signature BIT STRING,	M	
	certs [0] EXPLICIT SEQUENCE OF Certificate OPTIONAL }	O	
	AcceptableResponses ::= SEQUENCE OF { id OBJECT IDENTIFIER }	O	
	CrlID ::= SEQUENCE {	O	
	crlUrl [0] EXPLICIT IA5String OPTIONAL,	O	

SECTION	FEATURE		STATUS	REMARKS
	crlNum	[1] EXPLICIT INTEGER OPTIONAL,	O	
	crlTime	[2] EXPLICIT GeneralizedTime OPTIONAL }	O	
	-- Object Identifiers			
	id-pkix-ocsp	OBJECT IDENTIFIER ::= { id-ad-ocsp }	M	
	id-pkix-ocsp-basic	OBJECT IDENTIFIER ::= { id-pkix-ocsp 1 }	M	
	id-pkix-ocsp-nonce	OBJECT IDENTIFIER ::= { id-pkix-ocsp 2 }	M	
	id-pkix-ocsp-response	OBJECT IDENTIFIER ::= { id-pkix-ocsp 3 }	M	
	id-pkix-ocsp-crl	OBJECT IDENTIFIER ::= { id-pkix-ocsp 4 }	M	
	id-pkix-ocsp-signature	OBJECT IDENTIFIER ::= { id-pkix-ocsp 5 }	M	

Document Point of Contact:

Defense Information Systems Agency
ATTN: JIEO-JEBBC (Gregor D. Scott)
Ft. Monmouth, NJ 07703-5613
USA
Voice: 732-427-6856
Fax: 732-532-0853
Email: scottg@ftm.disa.mil